4. Release A Architecture Overview

4.1 Introduction

Sections 4 and 5 provide a hierarchical overview of the Release A architecture and design. Section 4 discusses the Release A context within the EOSDIS Ground System and its interfaces to all external systems as well as cross DAAC Interfaces. The section also discusses the partitioning of Release A into segment and subsystems. Section 5 continues this partitioning to software and hardware configuration items and local area network (LAN) design. Much of this material is abstracted from the subsequent detailed design documents (305-CD-005-001 through 305-CD-019-001) and is presented here only as an overview. These documents continue the system partitioning through the use of object models and associated material. The reader is referred to those documents for a more in-depth understanding of the Release A Detailed Design.

4.2 Release A Context Description

Release A provides capabilities to support the Mission Objectives described in Section 3. Release A objectives can be divided into two general classification: Operational Support for Current Missions and Interface Testing Support for Future Missions. These classifications provide a context to describe the capabilities and interfaces for ECS Release A at the EOSDIS Facilities.

The Release A external interfaces to ECS and between ECS components (SMC and DAACs) that support the Operational Support capabilities are presented in Figure 4.2-1. Release A Operational Support is provided for the TRMM spacecraft including interfaces to SDPF and TSDIS and full operational capability for the ingest and archival of TRMM data for CERES and LIS instruments. Product generation, receipt, archive and distribution of TRMM data products received from TSDIS, with metadata, ancillary data and the L1-L4 archive capabilities at the GSFC and MSFC and LIS and CERES capabilities at LaRC and MSFC. In addition Operational Support provides access to all V0 data including data still held in V0 data servers through V0 interoperability, and V0 data migrated to ECS data servers. Release A also provides V0 users access to all ECS held data, thus providing two way V0 interoperability.

Release A Interface Testing Support is provided for the EOS-AM-1 and LANDSAT-7 missions. For EOS-AM-1, Release A provides capabilities for early functional testing of EOS AM-1 ground system interfaces among several facilities including the Flight Dynamics Facility (FDF), the Aster Ground Data System, NOAA/NESDIS, the GDAO, the ECS Data Operation System (EDOS), and three DAACs - LaRC, GSFC, and EDC. Science Software Integration and Test (SSI&T) is also provided Version 1 of EOS-AM-1 instruments at LaRC, GSFC, and EDC. LANSAT-7 Release A also provides capabilities for early functional testing of Landsat-7 ground system interfaces between the Landsat Processing System and ECS at EDC. The Release A external interfaces to ECS and between ECS components (SMC and DAACs) that support the Interface Testing capabilities are presented in Figure 4.2-2.

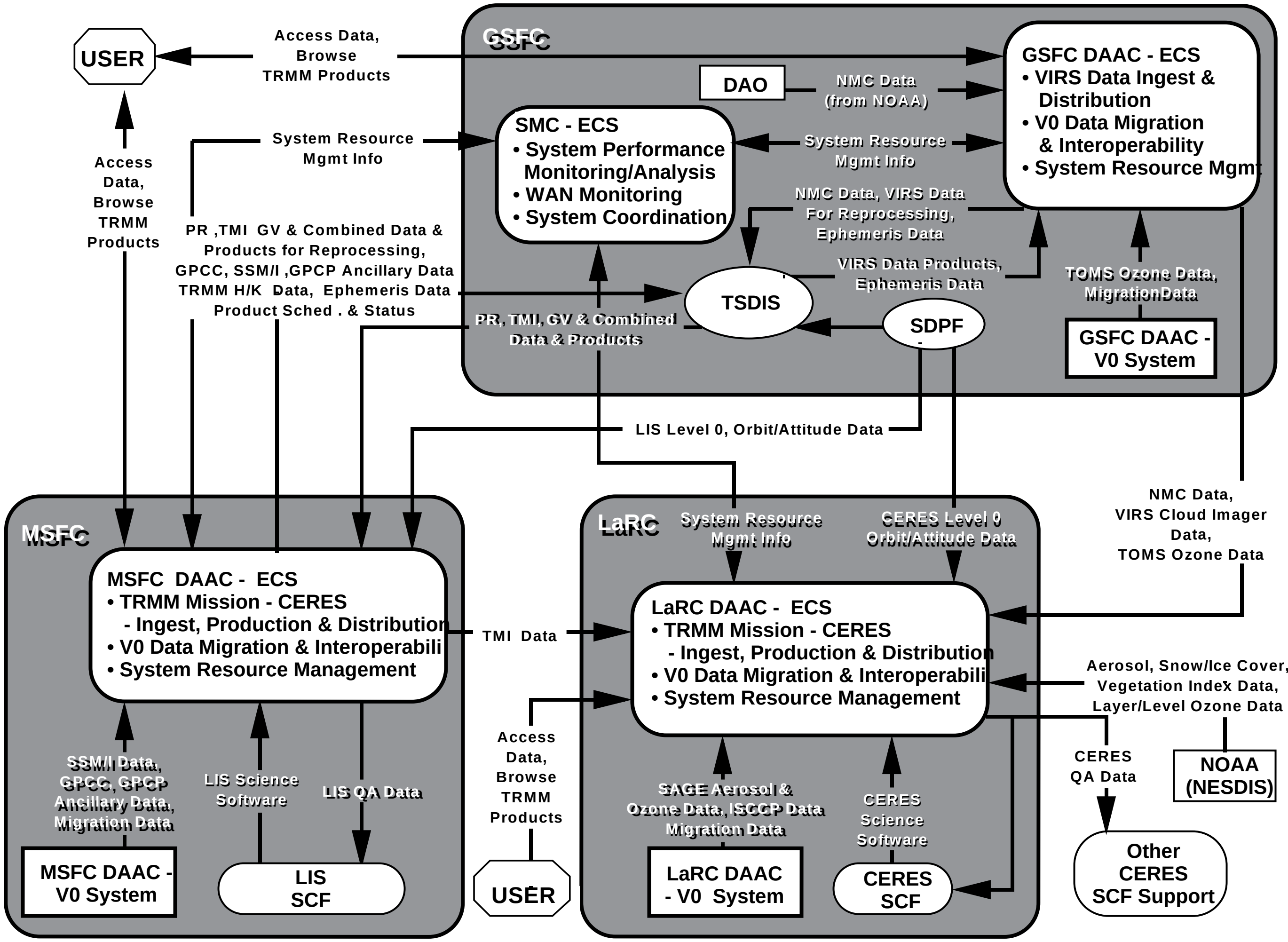

Figure 4.2-1 TRMM Mission Key Interfaces (Release A)

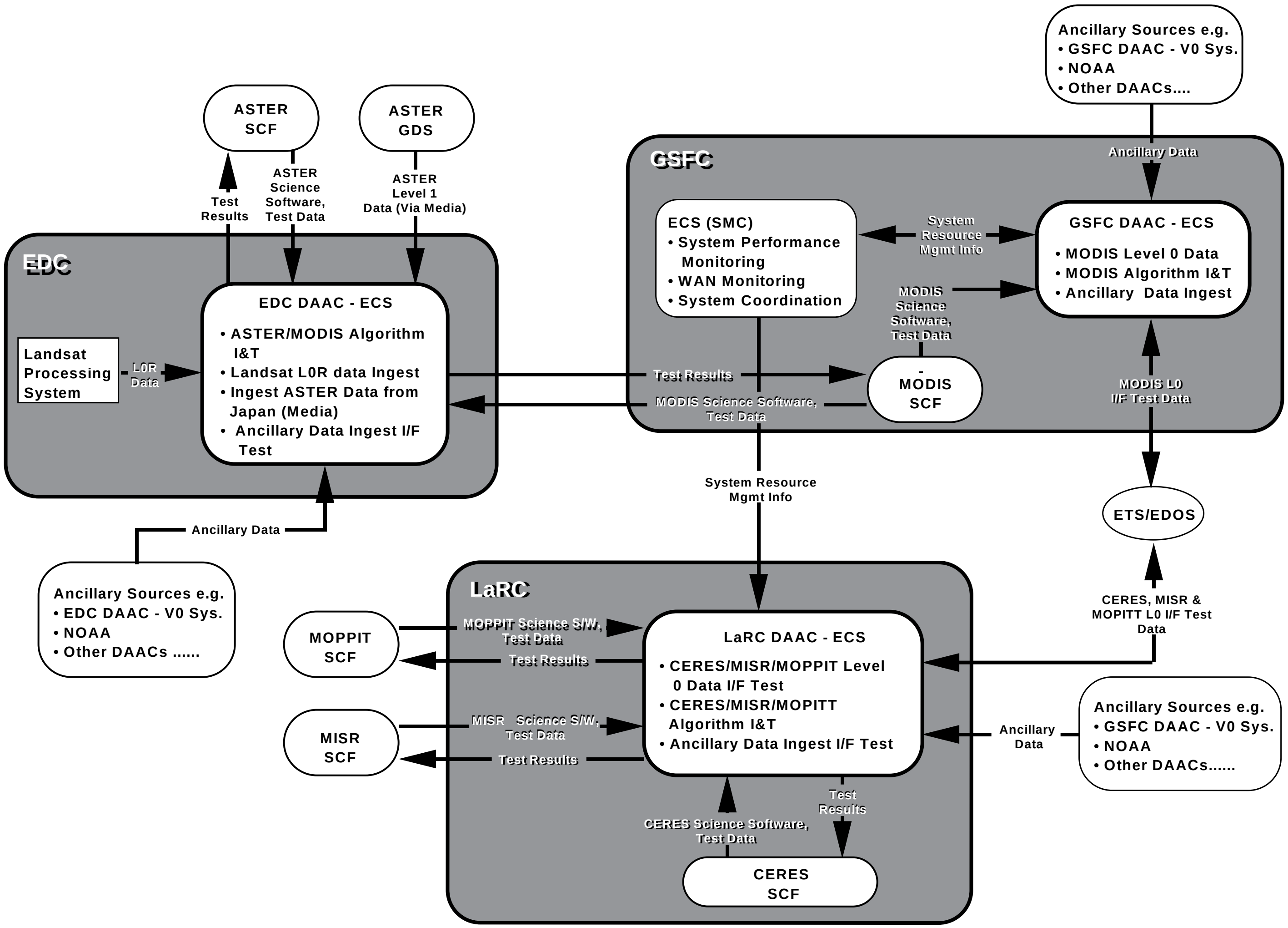

Figure 4.2-2 LANDSAT-7 amd AM-1 Key Interfaces (Release A)

These interface diagrams provide a context for understanding the Release A architecture and design components and their mission at each EOSDIS Facility. The remaining sections provide an overview of the Release A design and as such do not deal specifically with the configuration of components at each EOSDIS Facility. For more information on the site unique configurations refer to the DAAC Implementation subdocuments (305-CD-014-001 through 305-CD-019-001). Additional details on the Release A External Interfaces can be found in the Release A External Interface Control Documents listed in Section 2.

4.3 Release A Architecture

The ECS design has been partitioned into three segments (Figure 4.3-1), namely a:

- Science Data Processing Segment (SDPS) which is responsible for ECS application which provide
  - data management and archiving functions,
  - a processing environment for the execution of science software,
  - external interfaces for the acquisition of data needed for processing or intended for archiving, and
  - functions which support the search and retrieval of ECS managed data by science and other users.
- Communications and System Management Segment (CSMS) which is responsible for all communications, networking, and enterprise management functions, including
  - a distributed applications and operating system infrastructure,
  - various communications services such as electronic mail and file transfer,
  - monitoring and management of networking, system, and application resources,
  - access control and security management, and
  - local area network services and external network connectivity.
- Flight Operations Segment (FOS) which is responsible for space craft and instrument command and control functions.

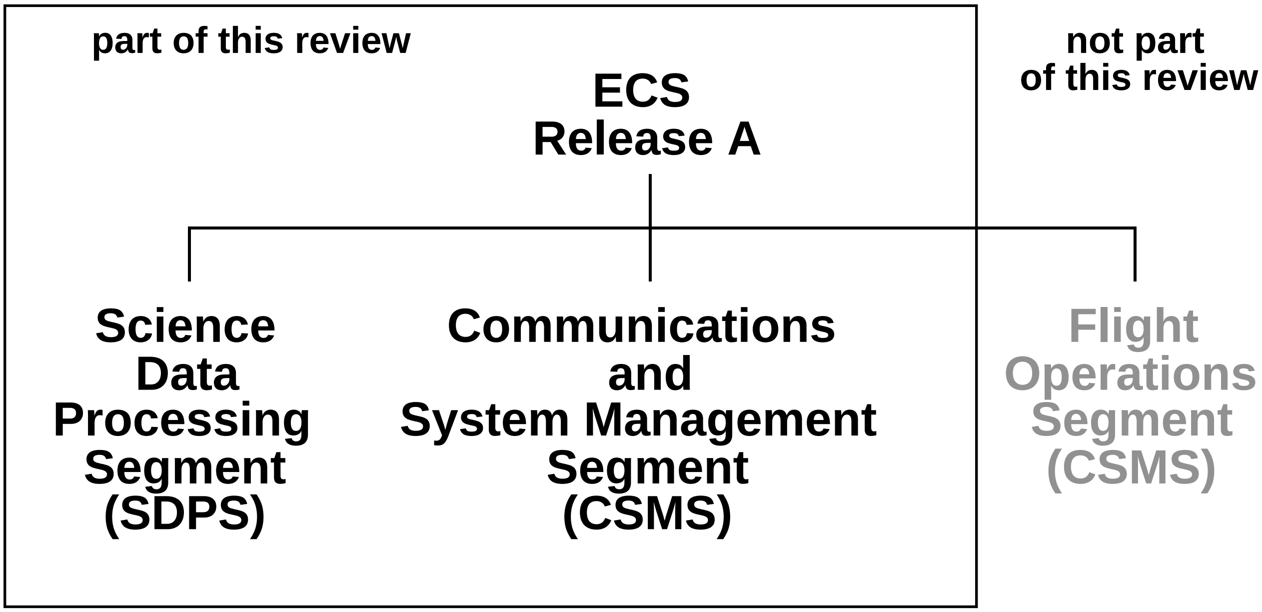

Figure 4.3-1. ECS Design Segments

This detailed design document addresses the Release A capabilities of Science Data Processing, and the Communications and System Management capabilities they require. It does not include FOS and its supporting CSMS functions, which are the subject of a separate Critical Design Review in the near future.

The following sub-sections provide an overview of the SDPS and CSMS subsystems. Section 5, which provides the formal breakdown of the segments below the level of subsystems, into software configuration components (CSCI) and hardware configuration components (HWCI), including the functions allocated to each and pointers to the subsystem detailed design subdocuments of DID 305 which describe the design of each of these components.

4.3.1 SDPS Segment Architecture

SDPS is composed of seven (7) subsystems. They provide the hardware and software resources needed to implement the SDPS functionality. The subsystems are defined in the SDS and have been presented at SDR. This section provides a brief review (see Figure 4.3-2).

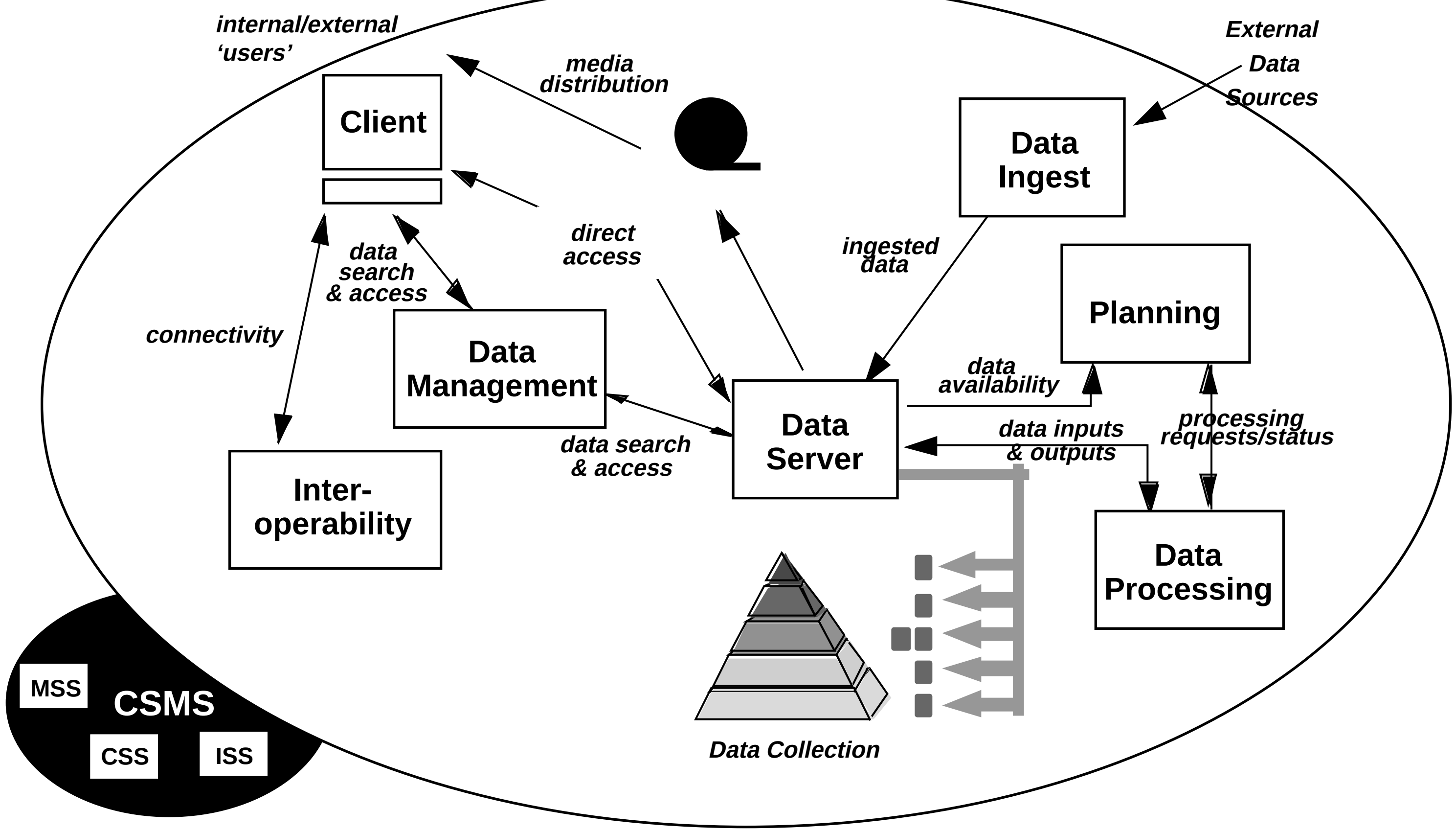

Figure 4.3-2 SDPS Subsystem Architecture

The subsystems can be grouped into the following four categories:

- *Data Storage and Management* - as represented by the Data Server Subsystem (DSS), provides the functions needed to archive science data, search for and retrieve archived data, manage the archives, and stage data resources needed as input to science software or resulting as output from their execution.

  A key driver in the architecture of this subsystem was that the costs for data storage and processing capacities are in rapid decline, hand-in-hand with improvements in throughput and response times. In the long term, therefore, ECS will be best served with an approach which affords the full capabilities of database management to science data, even though this approach cannot be fully implemented in the near future. Therefore, the Data Server Subsystem provides access to earth science data (which used to be divided into layers called a "data pyramid") in an integrated fashion through an Application Programming Interface that is common to all layers.

- *Data Search and Retrieval* - (also called the "Data Pull Side" of the system) is represented by the science user interface functions in the Client Subsystem (CLS), data search support functions in the Data Management Subsystem (DMS), and capabilities in the Interoperability Subsystem (IOS) which assist users in locating services and data of interest to them and their projects.

  The following were key drivers in the architecture of these subsystems. First, these capabilities should be capable of evolving to meet the needs of a wider GCDIS or UserDIS, as described in Section 3. Second, ECS should provide the basic underpinning for an earth science workbench, the assumption being that some of the science community will be in a much better position to develop the data viewing, search, and analysis tools they need which should fit into this workbench. Finally, as the Data Server Subsystem evolves to provide DBMS functionality for earth science data, the data pull subsystems should be able to undergo a similar evolution to support this concept for the end user.

- *Data Processing* - (considered a part of the "Data Push Side" of the system) is represented by a processing environment (the Data Processing Subsystem or DPS) for the science software; and capabilities for long and short term planning of science data processing, as well as management of the production environment provided by the Planning Subsystem (PLS).

  In the near term, data processing and re-processing will occur routinely and in accordance with the established production plans. However, long term ECS should be able to provide "production on demand", where higher level products are produced only when there is explicit demand for their creation. The ability to evolve to a mix of routine production and on-demand processing as users ask for data not currently stored by the Data Server Subsystem is a key driver in the subsystems' software architecture. The need to provide highly scalable and configurable processing resources is the main driver in their hardware architecture.

- *Data Ingest* - (also considered part of the "Data Push Side") is represented by the Ingest Subsystem (INS). The subsystem provides the interfaces with external applications, data staging capabilities, and storage for an approximately 1-year buffer of Level 0 data (so that reprocessing can be serviced from local storage).

The number of external interfaces which ECS will have is potentially very large, and the interfaces can serve very diverse functions, such as high-volume ingest of level 0 data and low-volume ingest of data from field campaigns. Thus, the ability to dynamically configure all or a large part of an external interface from standardized components is a key driver of the subsystem's software architecture. Resiliency against failure and loss of ingested data is a key driver in its hardware configuration.

The following sub-sections provide brief overviews for each of these subsystems. More detailed discussions of their design breakdown can be found in Section 5. Detailed design presentations are reserved for the corresponding DID 305 design subdocuments.

4.3.1.1 Client Subsystem (CLS)

The SDPS client subsystem has three main objectives:

- provide earth science users with an interface via which they can access ECS services and data
- offer an environment into which science users can integrate their own tools
- give science programs access to the ECS services, as well as direct access to ECS data

The client subsystem software, therefore, consists of graphic user interface (GUI) programs, tools for displaying the various kinds of ECS data (e.g., images, documents, tables), and libraries representing the client API of ECS services. Modern user interfaces are based on an object paradigm. The SDPS client subsystem is no exception; the graphic user interface programs will follow an object oriented design. The design will be built around a core set of 'root' objects from which all other GUI software will inherit its behavior. This will lead to a consistent look and feel and reduce the amount of software that needs to be developed. This core set is called the desktop. The remainder of the software is collectively called the workbench.

For Release A, the client subsystem will consist of the desktop, a user interface which allows users to search and browse a database describing the data and services available within ECS (called the advertising database), and a data visualization tool (EOSView). The remainder of the Release A user interface will be provided by an enhanced version of the V0 System Client (also referred to as the Release A Client). It provides data search and access for ECS science data, and a browsing interface for Guide documents and other types of ECS document data.

For the purposes of the SDPS design, it has been decided to group interfaces required by operational staff with the respective subsystem, and within each subsystem, with the respective CSCI. This ensures that operational interface requirements are properly addressed and presented in context. However, it may raise concern that there might be plethora of a user interface styles in SDPS. In fact, development of these interfaces (where they are not off the shelf) will be consolidated within a single Release A user interface development group, and will follow the guidelines developed by a team of Human Factors Engineering (HFE) experts. The guidelines are documented in the updated version of the ECS User Interface Style Guide, Version 5 and they are summarized in Section 6.5. The client subsystem components will be deployed on the hardware components which support operational staff, and provision for the necessary user interface hardware exists in the respective HWCI.

4.3.1.2 Interoperability Subsystem (IOS)

The SDPS is architected as a collection of distributed applications. They need support by distributed operating system and communications services. These are part of the CSMS and are described in the CSS Design Specification [305-CD-012-001]. To these functions, the SDPS interoperability subsystem adds an "advertising service." It maintains a database of information about the services and data offered by ECS, and provides interfaces for searching this database and for browsing through related information items. The Client Subsystem provides a user interface which enables scientists to locate services and data that may be of interest to them. This software is being developed in an incremental fashion, and early versions have been made available in Evaluation Packages (EP) to provide science user feedback to the developers.

Since Release A is of limited scope, it could be implemented and operated without an IOS. However, the interfaces and capabilities of the IOS play such an important part in the overall system concept, that it has been decided to include the IOS in the Release A SDPS. The advertising service will be implemented as an SDPS developed distributed database application on top of a commercial off-the-shelf Data Base Management System (Sybase).

4.3.1.3 Data Management Subsystem (DMS)

The subsystem provides three main functions:

- Provide a dispersed community of science users with services to search a set of data repositories (however, the repositories themselves and their search, access, and data management functions are part of another subsystem).
- Allow those scientists to obtain explanations for the data offered by these repositories.
- Provide data search and access gateways between ECS and external information systems.

The subsystem includes distributed search and retrieval functions and corresponding site interfaces, however, they are not part of the Release A design. Rather, Release A uses the capabilities of the Version 0 IMS to provide both the user search interface, and the intersite search functions. Nevertheless, the Release A design documents provide an overview of these functions as context for understanding the other data management functions.

In addition, the subsystem includes an interface between ECS and the Version 0. Release A implements the Version 0 to ECS gateway, in support of the Version 0 IMS user interface. The gateway accepts Version 0 requests (i.e., directory search, inventory search, browse, order and guide access), and translates them into access to the ECS Data Server Subsystem. The gateway also reformats the results of search and data access requests into the formats used by the Version 0 IMS. ECS gateways will follow a general gateway architecture to allow reuse of design components. This architecture is presented in Section 6.

4.3.1.4 Data Server Subsystem (DSS)

The subsystem provides the physical storage access and management functions for the ECS earth science data repositories. Other subsystems can access it directly or via the data management subsystem (if they need assistance with searches across several of these repositories). The subsystem also includes the capabilities needed to distribute bulk data via electronic file transfer or physical media. The main components of the subsystem are the following:

- Database Management System - SDPS will use an off-the-shelf DBMS (Sybase) to manage its earth science data and implement spatial searching, as well as for the more traditional types of data (e.g., system administrative and operational data). It will use a document management system to provide storage and information retrieval for guide documents, scientific articles, and other types of document data.
- File Storage Management Systems - they are used to provide archival and staging storage for large volumes of data. SDPS is considering the use of several hardware/software configurations which are either off-the-shelf or a mixture of off-the-shelf and developed software.
- Data Type Libraries - they will implement functionality of earth science and related data that is unique and not available off the shelf (e.g., spatial search algorithms and translations among coordinate systems). The libraries will interface with the data storage facilities, i.e., the database and file storage management systems.

Other components include, for example, administrative software to manage the subsystem resources and perform data administration functions (e.g., to maintain the database schema); and data distribution software, e.g., for media handling and format conversions.

The type library concept is at the heart of the DSS, and is the key mechanism via which the subsystem plans to achieve the long term goal of providing database management capabilities for earth science data. In analogy to a database management system, each data type is registered in the data server schema, which describes the capabilities provided by its type library. The type library then can be accessed via the Release A API (Release B will expand this approach to provide object-relational database access capabilities).

Although the Release A DSS needs to support only a limited number of earth science data types, the concept of a type-oriented data management system has far-reaching implications for its internal design. Therefore, the Release A DSS, though limited to the Release A data types, must implement most of the data server architecture presented at SDR. Similarly, though Release A data volumes and I/O throughput requirements are well below those predicted for Release B, the fact that Release B follows only nine (9) months after Release A requires that the Release A DSS design take the expansion in capacity required for Release B into account.

4.3.1.5 Ingest Subsystem (INS)

The subsystem deals with the initial reception of all data received at an EOSDIS facility and triggers subsequent archiving and processing of the data.

Given the variety of possible data formats and structures, each external interface, and each ad-hoc ingest task may have unique aspects. Therefore, the ingest subsystem is organized into a collection of software components (e.g., ingest management software, translation tools, media handling software) from which those required in a specific situation can be readily configured. The resultant configuration is called an ingest client. Ingest clients can operate on a continuous basis to serve a routine external interface; or they may exist only for the duration of a specific ad-hoc ingest task.

The ingest subsystem also standardizes on a number of possible application protocols for negotiating an ingest operation, either in response to an external notification, or by polling known data locations for requests and data. The subsystem will use the components of the general ECS external interface architecture which are presented in Section 6.

Special considerations exist with respect to the handling and archiving Level 0 data. This is a major and mission critical part of the SDPS interfaces and design (see SDPS Ingest Subsystem Design Specification [305-CD-009-001]).

4.3.1.6 Data Processing Subsystem (DPS)

The main components of the data processing subsystem - the science algorithms - will be provided by the science teams. The data processing subsystem will provide the necessary hardware resources, as well as a software environment for queuing, dispatching and managing the execution of these algorithms. Eventually, the processing environment will be highly distributed and will consist of heterogeneous computing platforms. Although this is not a driving consideration for Release A, it has had a tremendous impact on the resultant design. For example, ECS has selected a product (AutoSys) which is well suited for managing production in a distributed, heterogeneous UNIX environment, and Release A is integrating the product with custom software designed to manage distributed, heterogeneous computing resources.

The DPS also interacts with the DSS to cause the staging and de-staging of data resources in synchronization with processing requirements. The Release A processing environment will be much simpler than that of later releases, however, the DPS design must be able to accommodate the expansion to Release B which will occur only 9 months after Release begins operation. The consequences for the Release A design are discussed in SDPS Data Processing Subsystem Design Specification [305-CD-011-001].

4.3.1.7 Planning Subsystem (PLS)

The subsystem provides the functions needed to pre-plan routine data processing, schedule ad-hoc processing, and dispatch and manage processing requests. The subsystem provides access to the data production schedules at each site, and provides management functions for handling deviations from the schedule to operations and science users. The ECS processing environment poses a number of challenges:

- a processing environment which eventually will be highly distributed and consist of heterogeneous computing platforms
- existence of inter-site and external data dependencies
- dynamic nature of the data and processing requirements of science algorithms
- need for high availability

- providing a resource scheduling function which can accommodate hardware technology upgrades
- evolution to support substantial on-demand processing (as an alternative to predominantly routine processing) in future releases
- ability to provide longer-term (e.g., monthly) processing predictions as well as short term (e.g., daily) planning and scheduling

The complexity of and demands on the processing environment will increase dramatically with later Releases of SDPS. Release A, therefore, has been careful to select design approaches which will hold up in future releases. It was decided to separate long term planning as a custom software implementation, from short term planning and scheduling as provided by commercial off the shelf software (AutoSys). Production planners will be able to study the impact of changes in resource allocation or processing plans (e.g., the introduction of substantial reprocessing) on overall schedules. At the same time, operations personnel can make decisions regarding the daily workload in a timely fashion, interacting with a highly responsive database, and using a robust commercial product specifically designed for that purpose.

4.3.2 CSMS Segment Architecture

Release A has been designed as a fully distributed, heterogeneous system. To implement this, SDPS will make use of the services which are provided by Communications and System Management Segment (CSMS) (see Figure 4.3-3), most notably, its Communications Services Segment (CSS). The following are a few examples:

- SDPS internal interfaces typically use the CSMS Distributed Object Framework.
- A number of candidate COTS software components of SDPS are currently using UNIX sockets (e.g., Sybase), and will rely on that CSMS service.
- Many external interfaces as well as the bulk data transfers within SDPS will use CSMS file transfer services.
- SDPS will utilize the interfaces offered by MSS for fault and event reporting, and to obtain security management information.
- Some SDPS components will generate electronic mail messages (e.g., Science Data Server notices, and subscription notices sent via electronic mail).

In addition, a bulletin board service, electronic mail, file transfer and distributed file services, telnet, and other CSMS services which can be invoked directly by a user via a CSMS (or operating system) provided user interface will be used by science users and operational staff to perform various SDPS related functions (e.g., operations might notify a scientist of the arrival of a software delivery vie e-mail). These interfaces are not called out in the SDPS design, since they do not involve SDPS software components, but they are mentioned where appropriate in the SDPS scenarios.

SDPS relies extensively on the security management and authentication services provided by CSMS, and will add security services in areas where SDPS software components need to provide security for internally managed objects whose structure is transparent to CSMS (e.g., there will be additional security capabilities within the Data Server CSCIs).

The SDPS design also makes use of a universal method for referencing data objects, called the *Universal Reference* (UR). Universal References encapsulate the identifier of a data object, as well as a (location independent) network name of an SDPS service which can interpret the object identifier and access the object. The CSMS Directory/Naming Service will be responsible for providing and maintaining the network names of SDPS services.

The following sub-sections briefly describe the CSMS subsystems and characterize their relationships with one another, SDPS and FOS, and external entities discussed above. More detailed material is provided in the corresponding subsystem design documents of DID305.

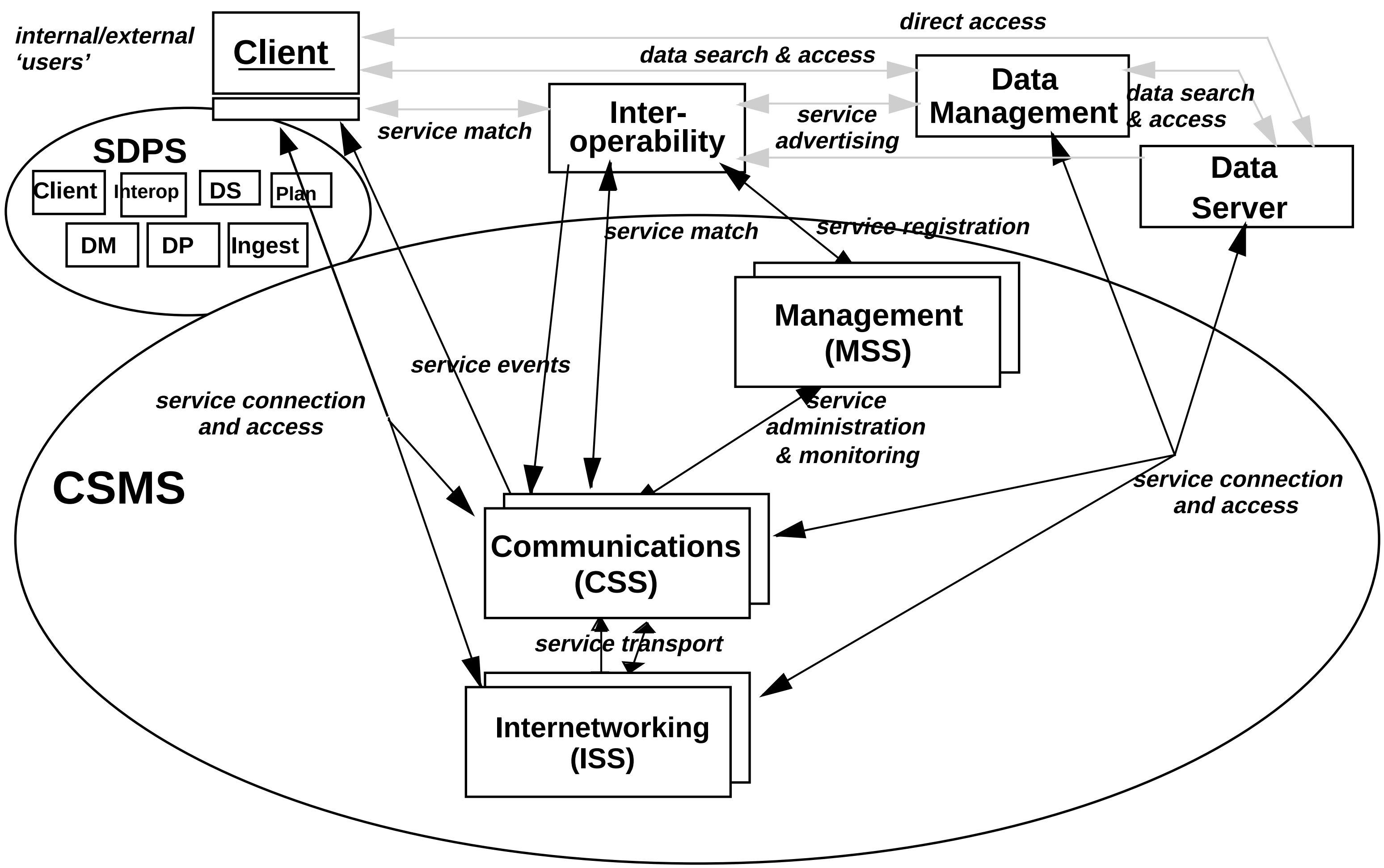

Figure 4.3-3 CSMS Subsystem Architecture

4.3.2.1 Communications Subsystem (CSS)

CSS plays a key role in the interoperation of the SDPS subsystems. SDPS applications follow an object-oriented design. That is, their lowest level software components are "software objects". SDPS also implements a distributed design, that is, its components - the software objects - are distributed across many platforms at a given site, and across several sites. For the software objects to communicate with each other requires "distributed object" communications environment. This environment is provided by CSS, using off-the-shelf technology (OO-DCE from Hewlett-Packard) augmented with some custom software. The environment allows software objects to communicate with each other reliably, synchronously as well as asynchronously, via interfaces which make the location of a software object and the specifics of the communications mechanisms transparent to the application.

In addition, CSS provides the infrastructural services for the distributed object environment. They are based on the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF). DCE includes a number of basic services needed to develop distributed applications, such as remote procedure calls (rpc), distributed file services (DFS), directory and naming services, security services, and time services.

Finally, CSS provides a set of common facilities, which include legacy communications services required within the ECS infrastructure and at the external interfaces for file transfer, electronic mail, bulletin board and remote terminal support. The Object Services support all ECS applications with interprocess communication and specialized infrastructural services such as security, directory, time, asynchronous message passing and event logging. The Distributed Object Framework is a collection of a set of core object services, collectively providing object-oriented client server development and interaction amongst applications.

4.3.2.2 Management Subsystem (MSS)

The Management Subsystem (MSS) provides enterprise management (network and system management) for all ECS resources: commercial hardware (including computers, peripherals, and network routing devices), commercial software, and custom applications. With few exceptions, the management services will be fully decentralized, such that no single point of failure exists which would preclude the system from continuing to operate or system operations and management to come to a halt.

However, MSS does provide two levels of an ECS management view: the local (site/DAAC specific) view is provided by Local System Management (LSM), and the enterprise view is provided by Enterprise Monitoring and Coordination (EMC) at the SMC, located at Goddard Space Flight Center (GSFC).

Enterprise management relies on the collection of information about the managed resources, and the ability to send notifications and commands to those resources. For network devices, computing platforms, and some commercial of the shelf software, MSS relies on software called "agents" which is usually located on the same device/platform and interacts with the device's or platform/s control software, or the commercial software product.

However, a large portion of the ECS applications software is custom developed, and some of this software - the science software - is externally supplied. For these components, MSS provides a set of interfaces via which these components can provide information to MSS (e.g., about events which are of interest to system management such as the receipt of a user request or the detection of a software failure), and or can take commands from MSS provided to MSS from M&O consoles (e.g., an instruction to shut down a particular component).

Applications which do not interact with MSS directly will be monitored by software which acts as their "proxies". For example, the Data Processing Subsystem (DPS) acts as the proxy for the science software it executes. It notifies MSS of events such as the dispatching or completion of a PGE, or its abnormal termination.

ECS selected HP OpenView as the centerpiece of its system management solution, and is augmenting it with other commercially available "agents", as well as custom developed software (e.g., the applications interfaces mentioned above). The information collected via the MSS interfaces from the various ECS resources is consolidated into an event history database on a regular basis (every 15-to 30 minutes) as well as on demand, when necessitated by an operator inquiry. The database is managed by Sybase, and Sybase query and report writing capabilities will be used to extract regular and ad-hoc reports from it. Extracts and summaries of this information will be further consolidated on a system wide basis by forwarding it to the SMC (also on a regular basis). These reporting capabilities and strategies are further discussed in Section 6.

MSS also provides other general system management functions, such as security management (providing administration of identifications, passwords, and profiles), and configuration management for ECS software, hardware, and documents.

4.3.2.3 Internetworking Subsystem (ISS)

The ISS is a layered stack of communications services corresponding to layers 1-4 of the Open Systems Interconnect Reference Model (OSI-RM). The ISS provides local area networking (LAN) services at ECS installations to interconnect and transport data among ECS resources. The ISS includes all components associated with LAN services including routing, switching, and cabling as well as network interface units and communications protocols within ECS resources.

The ISS also provides access services to link the ECS LAN services to Government-furnished wide-area networks (WANs), point-to-point links and institutional network services. Examples include the NASA Science Internet (NSI), Program Support Communications Network (PSCN), and various campus networks "adjoining" ECS installations. More detail of ISS is provided in Section 5 of this document.